



Heuristics for operational fixed job scheduling problems with working and spread time constraints

Deniz Türsel Eliyi^{a,*}, Meral Azizoglu^b

^a Izmir University of Economics, Department of Industrial Systems Engineering, Sakarya Cad., No: 156, Izmir 35330, Turkey

^b Middle East Technical University, Department of Industrial Engineering, ODTU 06531, Ankara, Turkey

ARTICLE INFO

Article history:

Received 12 October 2009

Accepted 15 March 2011

Available online 30 March 2011

Keywords:

Fixed job scheduling

Working time constraints

Spread time constraints

Heuristics

ABSTRACT

Operational fixed job scheduling problems select a set of jobs having fixed ready and processing times and schedule the selected jobs on parallel machines so as to maximize the total weight. In this study, we consider working time and spread time constrained versions of the operational fixed job scheduling problems. The working time constraints limit the total processing load on each machine. The spread time constraints limit the time between the start of the first job and the finish of the last job on each machine. For the working time constrained problem, we present a filtered beam search algorithm that evaluates the promising nodes of the branch and bound tree. For the spread time constrained problem we propose a two phase algorithm that defines the promising sets for the first jobs and finds a solution for each promising set. The results of our computational tests reveal that our heuristic algorithms perform very well in terms of both solution quality and time.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Fixed job scheduling (FJS) is an applied area of scheduling where tasks (jobs) with specified weights, ready times and deadlines are to be processed on parallel resources (machines). A task can be processed only between its ready time and deadline, and its processing time is exactly equal to the difference between these.

The FJS problems are of two types: operational and tactical. The tactical fixed job scheduling (TFJS) problems aim to minimize the number or cost of the resources while processing all tasks. The operational fixed job scheduling (OFJS) problems aim is to select a subset of tasks for processing so that the total weight is maximized.

The FJS problems have many application areas in manufacturing and service operations as mentioned in many reported work. Kovalyov et al. (2007) and Kolen et al. (2007) survey the potential applications of the FJS problems together with their theory. Bekki and Azizoglu (2008) study the OFJS problem on uniform parallel machines. Kroon (1990) and Kroon et al. (1995) mention the applications in assigning aircrafts to gates, and the capacity planning of maintenance personnel. Fischetti et al. (1987, 1992) and Martello and Toth (1986) consider an application of the TFJS

problem in scheduling the bus drivers whereas Wolfe and Sorensen (2000) consider an application of the OFJS problem in scheduling the earth-observing satellites. Other cited application areas include class scheduling (Kolen and Kroon, 1991), satellite data transmission (Faigle et al., 1999) and printed circuit board manufacturing (Spieksma, 1999).

The machine operating time constraints in FJS are introduced by Fischetti et al. (1987, 1989). These constraints are of two types: working time and spread time. Working time constraints impose a limit on the processing load of each machine, letting no machine operate for more than T time units. Spread time of a machine is defined as the time between the start of its first operation and the finish of its last operation, including the idle times in between. The spread time constraints limit this time span with an upper bound, S .

The operating time constraints find their applications in both manufacturing and service operations. In manufacturing environments, it may not be economical to operate some high-precision/high-cost equipment for more than a prespecified time (working time) and/or the machines may be available for a particular specified time period (spread time). In service environments like car rental systems, hotel room reservations and personnel scheduling, the resources may have operation time constraints imposed by the technical or union regulations.

Fischetti et al. show that the TFJS problems with working time and spread time constraints are strongly NP-hard, in their 1987 and 1989 studies, respectively. They develop lower bounds and branch and bound (B&B) algorithms and report on their

* Corresponding author. Tel.: +90 232 4888357; fax: +90 232 4888475.

E-mail addresses: deniz.eliyi@ieu.edu.tr (D.T. Eliyi), meral@ie.metu.edu.tr (M. Azizoglu).

satisfactory average-case performances based on the data generated close to the real-life situations. In their study, Fischetti et al. (1992) provide several polynomial time heuristic algorithms for both problems and report on their satisfactory worst-case performances.

The research on the OFJS problem with operating time constraints is very limited despite its practical importance. Bouzina and Emmons (1996) study the OFJS problem under working time constraints. The authors prove that the preemptive problem is polynomially solvable when all job weights are equal; however it is NP-hard in the ordinary sense when the job weights are arbitrary. The OFJS problems with spread time and working time constraints are studied by Eliiyi and Azizoglu in their 2006 and 2010 studies, respectively. They show that both problems are NP-hard in the strong sense and report their several polynomially solvable special cases. They propose B&B algorithms with several problem size reduction mechanisms and dominance conditions. For both problems, the algorithms return optimal solutions for problem instances with up to 100 jobs and 4 machines. Solyali and Ozpeynirci (2009) study the OFJS problem with spread time constraints, and develop a branch and price algorithm that outweighs the B&B algorithm proposed by Eliiyi and Azizoglu (2006). Very recently Rossi et al. (2010) develop two heuristic algorithms, namely greedy heuristic and genetic algorithm and report on the satisfactory behavior of their algorithms.

In this study, we consider the OFJS problem with working time and spread time constraints. For each problem, we develop several heuristic procedures that use efficient upper bounds and dominance conditions. Our aim is to find powerful lower bounds in reasonable times. There are two heuristic procedures that consider the spread time constraints (Rossi et al., 2010). We compare the performance of our heuristic algorithm with those procedures. Our computational study reveals that our algorithm outweighs greedy heuristic over all problem instances; however the genetic algorithm slightly outweighs our algorithm. To the best of our knowledge there is no reported heuristic study for the OFJS problem with working time constraints. As both problems have many practical applications and are to be solved quite frequently, our study fills an important gap in the literature and can be used for large problem size instances conveniently. The rest of the paper is organized as follows. We present the mathematical formulations of the problems in Section 2. We present our heuristic algorithms in Sections 3 and 4. Section 5 presents the results of our computational study designed to evaluate the performances of the algorithms. The conclusions are discussed in Section 6.

2. Problem formulations

There are n jobs to be scheduled for processing on m identical parallel machines. Job j arrives at its ready time r_j , and completes at its deadline d_j , if selected for processing. The processing time of job j is $p_j = d_j - r_j$ and its weight is w_j . Without loss of generality, we assume that all data are integer and the jobs are indexed in non-decreasing order of their ready times.

In the OFJS problem with working time constraints (OFJSW), each machine processes a maximum of T time units. We assume that the total processing time of all jobs, i.e., $\sum_{j=1}^n p_j$, is greater than T , so that the working time constraint is not redundant. For the OFJS problem with spread time constraints (OFJSS), recall that the spread time for machine k is defined as $(d_{k(l)} - r_{k(1)})$ where $k(1)$ is the first job and $k(l)$ is the last job processed by machine k . The spread time for machine k ($d_{k(l)} - r_{k(1)}$) cannot exceed its limit S in any feasible solution of the OFJSS problem.

The time is assumed to be divided into intervals not necessarily equal in length. We let $\{t_1, t_2, \dots, t_z\}$ be the sequence of the r_j and d_j values in chronological order with duplicates removed, and P_a be the set of available jobs in interval a , i.e., the set of jobs that resides interval $[t_a, t_{a+1})$ between its ready time and deadline, $a = 1, 2, \dots, z-1$.

The binary assignment variable is defined as follows:

$$x_{jk} = \begin{cases} 1, & \text{if job } j \text{ is processed on machine } k \\ 0, & \text{otherwise} \end{cases} \quad \forall j, k.$$

The following integer programming model formulated by Bouzina and Emmons (1996) represents the OFJSW problem:

$$\text{Maximize } \sum_{k=1}^m \sum_{j=1}^n w_j x_{jk} \quad (1)$$

subject to

$$\sum_{k=1}^m x_{jk} \leq 1 \quad j = 1, \dots, n \quad (2)$$

$$\sum_{j \in P_a} x_{jk} \leq 1 \quad k = 1, \dots, m \quad a = 1, 2, \dots, z-1 \quad (3)$$

$$\sum_{j=1}^n p_j x_{jk} \leq T \quad k = 1, \dots, m \quad (4)$$

$$x_{jk} \in \{0, 1\} \quad k = 1, \dots, m \quad j = 1, \dots, n \quad (5)$$

The objective function (1) maximizes the total weight of the processed jobs. Constraint set (2) ensures that each job is processed by at most one machine. The condition that no machine can process more than one job at a time is handled by constraint set (3). Working time constraints (T -constraints) are stated in constraint set (4). Constraint set (5) forces the integrality in job-machine assignments.

To impose the spread time limit, we define set I_j as the incompatibility set for job j , as $I_j = \{i \mid i > j; r_i < d_j \text{ or } d_i - r_j > S\}$. Then, the assignment of job j and any job $i \in I_j$ to the same machine violates the spread time constraint. The following integer programming model formulated by Eliiyi and Azizoglu (2006) represents the OFJSS problem:

$$\text{Maximize } \sum_{k=1}^m \sum_{j=1}^n w_j x_{jk} \quad (6)$$

subject to

$$\sum_{k=1}^m x_{jk} \leq 1 \quad j = 1, \dots, n \quad (7)$$

$$\sum_{j \in P_a} x_{jk} \leq 1 \quad k = 1, \dots, m \quad a = 1, 2, \dots, z-1 \quad (8)$$

$$x_{ik} + x_{jk} \leq 1 \quad k = 1, \dots, m, \quad j = 1, \dots, n-1, \quad i \in I_j \quad (9)$$

$$x_{jk} \in \{0, 1\} \quad k = 1, \dots, m \quad j = 1, \dots, n \quad (10)$$

The objective function (6) maximizes the total weight of the processed jobs. Constraint sets (7) and (8) ensure that each job is processed on at most one machine and no machine can process more than one job at a time, respectively. Constraint set (9) imposes the spread time constraints. Constraint set (10) forces the integrality in job-machine assignments.

3. A filtered beam search for the OFJSW problem

Beam search is a heuristic B&B algorithm, which evaluates nodes at each level, keeping the most promising β nodes while discarding the rest. β is called the beam width. As some nodes are

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات